

012 0743 U300

日本国特許庁
PATENT OFFICE
JAPANESE GOVERNMENT

JC872 U.S. PTO
09/863499
05/23/01

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日
Date of Application:

2000年 5月25日

出願番号
Application Number:

特願2000-155010

願人
Applicant(s):

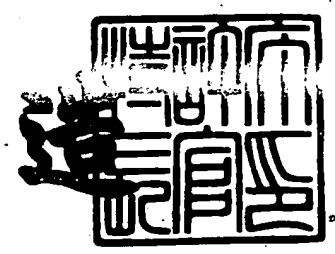
ソニー株式会社

CERTIFIED COPY OF
PRIORITY DOCUMENT

2001年 3月 9日

特許庁長官
Commissioner,
Patent Office

及川耕造



【書類名】 特許願

【整理番号】 0000483503

【提出日】 平成12年 5月25日

【あて先】 特許庁長官 近藤 隆彦 殿

【国際特許分類】 G06F 9/00

【発明者】

【住所又は居所】 東京都品川区北品川 6 丁目 7 番 3 5 号 ソニー株式会社
内

【氏名】 喜多山 卓郎

【特許出願人】

【識別番号】 000002185

【氏名又は名称】 ソニー株式会社

【代表者】 出井 伸之

【代理人】

【識別番号】 100082762

【弁理士】

【氏名又は名称】 杉浦 正知

【電話番号】 03-3980-0339

【手数料の表示】

【予納台帳番号】 043812

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9708843

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 コンポーネントウェア作成方法および記録媒体

【特許請求の範囲】

【請求項 1】 第 1 のコンポーネントウェアと、第 1 のコンポーネントウェアの参照元となる第 2 のコンポーネントウェアとのリンクを、システムにロードされたときに行うようにされたコンポーネントウェアを作成するコンポーネントウェア作成方法において、

外部から参照されるシンボル名を該シンボル名を識別するシンボル ID に置き換えて第 1 のコンポーネントウェアを生成する参照先コンポーネントウェア生成のステップと、

上記第 1 のコンポーネントウェアに対して参照元となる第 2 のコンポーネントウェア中の外部に参照するシンボル名を、該シンボル名を識別するシンボル ID に置き換えて上記第 2 のコンポーネントウェアを生成する参照元コンポーネントウェア生成のステップと

を有し、

上記シンボル ID に基づき上記第 1 および第 2 のコンポーネントウェアをリンクするようにしたことを特徴とするコンポーネントウェア作成方法。

【請求項 2】 請求項 1 に記載のコンポーネントウェア作成方法において、

上記第 1 のコンポーネントウェアには、該第 1 のコンポーネントウェアを識別するためのコンポーネント ID が付され、上記第 2 のコンポーネントウェアは、上記コンポーネント ID と上記シンボル ID とに基づき上記第 1 のコンポーネントウェアを参照し、上記リンクを行うようにしたことを特徴とするコンポーネントウェア作成方法。

【請求項 3】 請求項 1 に記載のコンポーネントウェア作成方法において、

上記参照先コンポーネントウェア生成のステップで上記第 1 のコンポーネントウェアを上記生成する際に、上記シンボル名と上記シンボル ID との対応関係を示すリストを作成するリスト作成のステップをさらに有し、

上記参照元コンポーネントウェア生成のステップで上記第 2 のコンポーネントウェアを上記生成する際は、上記リスト作成のステップで作成された上記リスト

を用いて、上記シンボル名を上記シンボル I D に置き換えるようにしたことを特徴とするコンポーネントウェア作成方法。

【請求項 4】 請求項 3 に記載のコンポーネントウェア作成方法において、
上記第 1 のコンポーネントウェアを再び生成する際に、上記シンボル名を削除した場合には、上記削除された上記シンボル名に対応する上記シンボル I D が上記削除された旨を上記リストに記録するようにし、新たなシンボルを追加する場合には、上記リストに基づき該新たなシンボルに対し、新たなシンボル I D を付するようにしたことを特徴とするコンポーネントウェア作成方法。

【請求項 5】 システムにロードされた際に外部のコンポーネントウェアから参照されリンクされるコンポーネントウェアが記録された記録媒体において、
外部から参照されるシンボル名が該シンボル名を識別するシンボル I D に置き換えられ、上記シンボル I D が該シンボル I D と対応するシンボルの参照場所情報と関連付けられて記述されると共に、コンポーネントウェア自身を識別するコンポーネント I D が記述されたシンボル情報テーブルを有するコンポーネントウェアが記録されたことを特徴とする記録媒体。

【請求項 6】 システムにロードされた外部のコンポーネントウェアを参照してリンクするコンポーネントウェアが記録された記録媒体において、

参照先のコンポーネントウェアに付された識別情報であるコンポーネント I D と、上記参照先のコンポーネントウェア内で参照されるシンボルの識別情報であるシンボル I D とが対とされて記述されると共に、上記コンポーネント I D で示される上記参照先のコンポーネントウェア内の上記シンボル I D で示される上記参照されるシンボルが上記参照先のコンポーネントウェア内で参照される参照場所情報が、上記コンポーネント I D と上記シンボル I D との対に対応付けられたシンボル参照情報テーブルを有するコンポーネントウェアが記録されたことを特徴とする記録媒体。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

この発明は、他のコンポーネントウェアのリンクをコンポーネントウェアのロ

ード時に動的に行うシステムにおけるリンク処理を効率的に行うことができるようにしたコンポーネントウェア作成方法および記録媒体に関する。

【 0 0 0 2 】

【従来の技術】

近年では、コンピュータ装置や携帯用端末装置、ディジタル放送を受信するためのセットトップボックスといった情報機器において用いられるソフトウェアは、コンポーネントウェア（以下、コンポーネントと略称する）と称される、部品として用いられるソフトウェアが組み合わされて構成される場合が多い。コンポーネントは、例えばソフトウェアプログラムが起動される際に必要に応じてロードされ、メモリ上に読み込まれることで当該ソフトウェアに組み込まれて使用される。

【 0 0 0 3 】

メモリ上に読み込まれたコンポーネントは、互いにリンクされて一つのソフトウェアを構成する。コンポーネントのリンクは、コンポーネント中に記述されているシンボル情報に基づき行われる。シンボル情報は、コンポーネントの作成時に、当該コンポーネントのソースファイル中の宣言に基づき、コンパイラによって自動的に設定される。シンボル情報は、シンボル情報を識別するためのシンボル名と、そのシンボル名に対応する相対的なアドレス情報であるシンボル値とからなる。相対的なアドレス情報は、例えば、ソースファイルがコンパイラでコンパイルされて生成されるオブジェクトファイル内でのアドレスのオフセット値である。シンボル名は、例えば、十数文字の英数字および記号の組み合わせにより表される。

【 0 0 0 4 】

例えばコンポーネントは、複数の関数を含み、関数のそれぞれに対してシンボル情報が設定される。シンボル情報は、コンポーネントのソースファイルがコンパイルされたオブジェクトファイルにおいて、シンボル情報テーブルとして管理される。実行ファイルの作成時には、リンクにより、リンク元のコンポーネントに記述されたシンボル情報に基づき、リンク先のコンポーネント上のシンボル情報テーブルが参照され、コンポーネントのリンクが行われる。実行時には、リン

ク元のシンボル情報に基づきリンク先から対応する関数が呼び出される。

【0005】

【発明が解決しようとする課題】

プログラムのロード時にリンクを行う場合、リンク元のコンポーネントでの宣言に基づくシンボル情報がリンク先のコンポーネントに存在するかどうかを調べる必要がある。これは、上述した、リンク先のコンポーネントのシンボル情報テーブルを参照して、所望のシンボル情報の検索を行うことでなされる。

【0006】

検索対象とされる外部シンボルの数が少ない場合には、短時間で処理が可能である。しかしながら、通常のライブラリなどでは、多くのシンボルを有し、検索に長い時間を要するという問題点があった。しかも、目的のシンボルの検索には、何度もシンボル名の比較を行う必要があり、長いシンボル名を解決するためには、より長い時間が必要となるといった問題点があった。

【0007】

また、シンボル名の比較照合を行う際には、検索対象となるシンボル名をメモリ上に展開する必要があり、大きなメモリ容量が必要となるといった問題点があった。

【0008】

さらに、同一のシンボル名を持つ二つ以上の異なるコンポーネントが同一のシステム上にロードされた場合、シンボル名の衝突が発生する。この場合、どのコンポーネントのシンボルが優先されるかは、ロードされた順番に依存するか、あるいは、ロードエラーとなってしまうという問題点があった。

【0009】

一方、このようなシンボル名の衝突を解決するために、例えば特許番号第2655612号では、ロードされた異なるコンポーネントに同一のシンボル名が存在した場合、従来は時間的に前にロードされたコンポーネントに定義されているシンボルが優先されていたものを、新たにロードされたコンポーネントのシンボルを参照することが可能とされている。しかしながら、この特許番号第2655612号に記載の方法では、例えば互いに別の開発者によって作られた二つのコ

ンポーネントに同一のシンボル名が存在した場合でも、同一のシンボルとしてしまうため、正しい動作を期待することができないという問題点があった。

【0010】

したがって、この発明の目的は、コンポーネントのリンクに要する処理時間を短縮すると共に、リンク時のメモリの使用量が少なく済むようなコンポーネントウェア作成方法および記録媒体を提供することにある。

【0011】

また、この発明の他の目的は、リンク関係にある二つ以上のコンポーネント間でのシンボル名の衝突を回避できるようなコンポーネントウェア作成方法および記録媒体を提供することにある。

【0012】

【課題を解決するための手段】

この発明は、上述した課題を解決するために、第1のコンポーネントウェアと、第1のコンポーネントウェアの参照元となる第2のコンポーネントウェアとのリンクを、システムにロードされたときに行うようにされたコンポーネントウェアを作成するコンポーネントウェア作成方法において、外部から参照されるシンボル名をシンボル名を識別するシンボルIDに置き換えて第1のコンポーネントウェアを生成する参照先コンポーネントウェア生成のステップと、第1のコンポーネントウェアに対して参照元となる第2のコンポーネントウェア中の外部に参照するシンボル名を、シンボル名を識別するシンボルIDに置き換えて第2のコンポーネントウェアを生成する参照元コンポーネントウェア生成のステップとを有し、シンボルIDに基づき第1および第2のコンポーネントウェアをリンクするようにしたことを特徴とするコンポーネントウェア作成方法である。

【0013】

また、この発明は、システムにロードされた際に外部のコンポーネントウェアから参照されリンクされるコンポーネントウェアが記録された記録媒体において、外部から参照されるシンボル名がシンボル名を識別するシンボルIDに置き換えられ、シンボルIDがシンボルIDと対応するシンボルの参照場所情報と関連付けられて記述されると共に、コンポーネントウェア自身を識別するコンポーネ

ントIDが記述されたシンボル情報テーブルを有するコンポーネントウェアが記録されたことを特徴とする記録媒体である。

【0014】

また、この発明は、システムにロードされた外部のコンポーネントウェアを参照してリンクするコンポーネントウェアが記録された記録媒体において、参照先のコンポーネントウェアに付された識別情報であるコンポーネントIDと、参照先のコンポーネントウェア内で参照されるシンボルの識別情報であるシンボルIDとが対とされて記述されると共に、コンポーネントIDで示される参照先のコンポーネントウェア内のシンボルIDで示される参照されるシンボルが参照先のコンポーネントウェア内で参照される参照場所情報が、コンポーネントIDとシンボルIDとの対に対応付けられたシンボル参照情報テーブルを有するコンポーネントウェアが記録されたことを特徴とする記録媒体である。

【0015】

上述したように、請求項1に記載の発明は、参照先である第1のコンポーネントウェアは、外部から参照されるシンボル名をシンボル名を識別するシンボルIDに置き換えて生成され、参照元である第2のコンポーネントウェアは、外部に参照するシンボル名を、シンボル名を識別するシンボルIDに置き換えて生成され、シンボルIDに基づき第1および第2のコンポーネントウェアをリンクするようにしているため、シンボルIDを短く設定すれば、シンボル名の照合の時間が短縮され第1および第2のコンポーネントウェアのリンク処理を高速化することができると共に、シンボル名の照合に必要なメモリの使用量を少なく抑えることができる。

【0016】

また、請求項5に記載の発明によれば、外部から参照されるシンボル名がシンボル名を識別するシンボルIDに置き換えられ、シンボルIDがシンボルIDと対応するシンボルの参照場所情報と関連付けられて記述されると共に、コンポーネントウェア自身を識別するコンポーネントIDが記述されたシンボル情報テーブルを有するコンポーネントウェアが記録媒体に記録されるため、記録媒体から読み出されたコンポーネントウェアは、他のコンポーネントウェアからシンボル

I Dで以てシンボルが参照されたときに、シンボル情報テーブルに基づきシンボルの参照場所を知ることができる。

【 0 0 1 7 】

また、請求項6に記載の発明によれば、参照先のコンポーネントウェアに付された識別情報であるコンポーネントI Dと、参照先のコンポーネントウェア内で参照されるシンボルの識別情報であるシンボルI Dとが対とされて記述されると共に、コンポーネントI Dで示される参照先のコンポーネントウェア内のシンボルI Dで示される参照されるシンボルが参照先のコンポーネントウェア内で参照される参照場所情報が、コンポーネントI DとシンボルI Dとの対に対応付けられたシンボル参照情報テーブルを有するコンポーネントウェアが記録媒体に記録されるため、記録媒体から読み出されたコンポーネントウェアは、他のコンポーネントウェアを参照するときに、シンボル参照情報テーブルに基づき他のコンポーネントウェア内の参照するシンボルの参照場所を知ることができる。

【 0 0 1 8 】

【発明の実施の形態】

以下、この発明の実施の一形態について、図面を参照しながら説明する。図1は、この発明の実施の一形態に適用可能なシステムの概念を示す。例えば携帯用情報機器やコンピュータ装置といった情報機器上に構成されるターゲットシステム1において、所定のソフトウェアの動作を実現するために、互いに関数などの呼び出しを行う関係にある複数のコンポーネント4がロードされる。

【 0 0 1 9 】

コンポーネントは、他のプログラムの部品として用いることができるプログラムを指す。コンポーネントは、例えば複数の関数を有し、他のコンポーネントとリンクされることで、他のコンポーネントやプログラムから関数の呼び出しや変数の参照などがなされる。以下では、説明のため、コンポーネント4Aの記述に基づき、コンポーネント4B内に定義されている関数や変数の参照が行われるものとする。すなわち、コンポーネント4Aがリンク元、コンポーネント4Bがリンク先とされ、コンポーネント4Aに対してコンポーネント4Bがリンクされる。図1の例では、リンク元のコンポーネント4A中の記述「func()」により、リ

ンク先である他のコンポーネント 4 B が参照され、コンポーネント 4 B が有する関数 func() が呼び出される。

【 0 0 2 0 】

ターゲットシステム 1 にロードされていないコンポーネント 4 は、例えばターゲットシステム 1 に接続された 2 次記憶装置 3 からロードすることができる（コンポーネント 4 C）。2 次記憶装置 3 としては、ハードディスク装置や CD-R OM (Compact Disc-Read Only Memory)、あるいは、その他の記録媒体を適用することができる。また、これに限らず、フラッシュメモリなどの半導体記憶素子を 2 次記憶装置 3 として適用させることができる。さらに、フロッピーディスクや MO (Magneto Optical Disk) などを 2 次記憶装置 3 として適用することができる。

【 0 0 2 1 】

ターゲットシステム 1 により、コンポーネント 4 A によって指定されるコンポーネント 4 B が 2 次記憶装置 3 から読み出される。読み出されたコンポーネント 4 B は、メモリに展開され、ターゲットシステム 1 にロードされる。

【 0 0 2 2 】

これに限らず、コンポーネント 4 は、ターゲットシステム 1 に接続されたネットワーク 2 を介して、サーバシステム 5 0 からダウンロードして、ターゲットシステム 1 にロードすることもできる（コンポーネント 4 D）。

【 0 0 2 3 】

図 2 は、ターゲットシステム 1 が構成される情報機器の一例の構成を示す。ここでは、ターゲットシステム 1 が携帯用情報機器 1 0 上に構成されるものとして説明する。勿論、ターゲットシステム 1 は、これに限らず、パーソナルコンピュータやセットトップボックス、携帯電話装置、家庭用ゲーム装置など、他の情報機器に構成されるようにしてもよい。バス 1 1 に対して CPU (Central Processing Unit) 1 2、RAM (Random Access Memory) 1 3、ROM (Read Only Memory) 1 3 および上述の 2 次記憶装置 3 が接続される。RAM 1 3 は、CPU 1 2 のワークメモリである。ROM 1 4 には、この携帯用情報機器 1 0 を動作させる基本的なソフトウェアである OS (Operating System) や、コンポーネントをロード

するためのコンポーネントローダ、コンポーネント同士のリンクを行うリンカ、起動時に必要とされるプログラムといった、その他のプログラムが予め記憶される。

【0024】

2次記憶装置3としては、例えば不揮発性で且つ書き替えが可能なメモリであるフラッシュメモリを用いることができる。この場合、2次記憶装置3は、換装可能とすることができる。2次記憶装置3は、これに限らず、ハードディスク装置を用いることができる。また、所定のドライブ装置を用いてMOやフロッピーディスクなどの記録媒体を2次記憶装置3として用いることができる。

【0025】

バス11に対して、さらに、表示部16、入力部17および通信インターフェイス18とが接続される。表示部16は、例えばLCD(Liquid Crystal Display)からなる。CPU12から表示制御コマンドが発行され、この表示制御コマンドに基づき所定の表示が表示部16になされる。入力部17は、例えば表示部16上に設けられ、表示部16の表示が透過可能なタッチパネルからなる。通信インターフェイス18は、無線あるいは有線通信によるネットワーク2と接続され、この携帯用情報機器10と外部との通信を制御する。

【0026】

携帯用情報機器10が起動されると、CPU12によりROM14に格納されたOSなどのプログラムが起動され、ターゲットシステム1が構成される。例えば入力部17からなされた指示により、ターゲットシステム1上で所定のプログラムが起動され、プログラム中の記述に従い、必要なコンポーネントが2次記憶装置3からRAM13に読み込まれ、ロードされる。

【0027】

コンポーネントにおけるリンクの指定は、コンポーネントの作成時になされる。リンクは、リンク元とリンク先とで対応するシンボルを参照することでなされる。この発明では、リンク先となるコンポーネントは、当該コンポーネントを示すコンポーネントIDを付されると共に、当該コンポーネントに含まれるシンボル情報が、例えば通し番号からなるシンボルIDで置き換えられた、シンボル情

報テーブルを有する。

【 0 0 2 8 】

一方、リンク元のコンポーネントは、リンク先となるコンポーネントのコンポーネント ID と、そのコンポーネント ID で示されるコンポーネント中の参照するシンボルのシンボル ID とが対とされる。そして、このコンポーネント ID とシンボル ID との対と、シンボル ID で示されるシンボルの、リンク先コンポーネント中での相対的なアドレスとが対応付けられた、シンボル参照情報テーブルを有する。

【 0 0 2 9 】

リンク元のコンポーネントは、シンボル参照情報テーブルに基づき、リンク先のコンポーネントおよびそのコンポーネント中で参照されるシンボルの相対的な位置を知ることができる。シンボル ID は、一般的に、オリジナルのシンボル名よりも短く設定することができる。このため、オリジナルのシンボル名がより短いシンボル ID で置き換えられるので、リンク先の検索が高速に行えると共に、メモリの使用量が抑えられる。

【 0 0 3 0 】

図 3 ～図 5 を用いて、この実施の一形態におけるコンポーネントの生成方法について説明する。なお、以下に説明するコンポーネントの生成は、ターゲットシステム 1 とは異なるシステムで実行することができる。

【 0 0 3 1 】

図 3 は、リンク先となるコンポーネント（参照先コンポーネント 4 B）の一例の生成方法を示す。リンク先となるコンポーネントは、外部から参照されるシンボルを持つ。図 3 A は、ソースファイル 1 0 において宣言された関数に対してシンボルが宣言された様子を示している。このソースファイルをコンパイラによってコンパイルすることで、図 3 B に示されるオブジェクトファイル 1 1 が生成される。オブジェクトファイルは、ソースファイルから実行型のファイルを作成する途中で生成される中間ファイルである。

【 0 0 3 2 】

ソースファイル 1 0 中で宣言された関数や変数で、他のコンポーネントから参

照されるシンボルは、オブジェクトファイル 11 中のシンボル情報テーブル 12 に列挙される。このとき、ソースファイル 10 においてなされたシンボルの宣言は、コンパイラによってシンボル名に置き換えられる。ここでは、シンボル情報テーブル 12 に書かれているシンボル名は、ソースコード 10 における記述に基づき生成された、オリジナルのシンボル名が用いられる。

【0033】

図 3 の例では、シンボル名は、「_func1_6Class1P6Class1」というような、十数文字の英数字および記号の組み合わせからなる文字列になっている。シンボル名には、宣言されたシンボルのオブジェクトファイル 11 における相対的なアドレス（オフセット値）を示すシンボル値が付される。これら、シンボルの宣言のシンボル名への置き換えとシンボル値の付加は、コンパイラによって自動的になされる処理である。

【0034】

このオブジェクトファイル 11 から、リンク先となる参照先コンポーネント 4B が作成される。このとき、この発明の実施の一形態では、オブジェクトファイル 11 中のシンボル情報テーブル 12 において、図 3C に一例が示されるように、シンボル名をシンボル ID に置き換えた、シンボル情報テーブル 14 が作成される。シンボル ID は、当該コンポーネント中で他のシンボル ID と重複しないような、任意の文字列とされる。図 3C の例では、上述のシンボル名「_func1_6Class1P6Class1」に対して「S3」がシンボル ID として設定される。シンボル名がシンボル ID に置き換えられたシンボル情報テーブル 14 には、さらに、当該コンポーネント 4B を示すコンポーネント ID（この例では、「C45」）が設定される。コンポーネント ID は、コンポーネント毎に一意に割り当てられる。

【0035】

また、この実施の一形態においては、コンポーネントファイル生成時に、作成されるコンポーネントファイルとは別に、外部シンボル情報ファイル 13 が作成される。外部シンボル情報ファイル 13 は、図 4 に一例が示されるように、コンポーネント ID に示されるコンポーネントについて、オリジナルのシンボル名と

シンボル ID との対応関係が各エントリ（行）に列挙される。

【 0 0 3 6 】

さらに、コンポーネント 4 B がバージョンアップなどにより再作成されシンボルが削除された場合、図 3 C に「DELETE」の行で示されるように、外部シンボル情報ファイル 1 3 に対して、シンボルが削除された旨が削除されたシンボル ID と共に記述される。このコンポーネントに対して新たにシンボルが追加される場合には、削除されたシンボルを含めて、過去に使われていないシンボル ID が使用される。このようにすることで、過去に使用されたシンボル ID が再使用されるのが防がれ、同一名で互いにバージョンの異なる複数のコンポーネントにおいて、シンボル ID の唯一性が保証される。

【 0 0 3 7 】

図 5 は、リンク元となるコンポーネント（参照元コンポーネント 4 A）の一例の生成方法を示す。リンク元となるコンポーネント 4 A は、外部のコンポーネントにおいて定義されている関数の呼び出しや、変数の参照を行う。図 5 A は、ソースファイル 2 0 において他のコンポーネントの関数が参照されることが示されている。このソースファイルがコンパイラによってコンパイルされ、図 5 B に示されるオブジェクトファイル 2 1 が生成される。

【 0 0 3 8 】

オブジェクトファイル 2 1 は、シンボル参照情報テーブル 2 2 を有する。シンボル参照情報テーブル 2 2 は、外部のコンポーネントに参照するシンボルが列挙される。このときの参照先のシンボル名は、ソースファイル 2 0 における参照を指示する記述に基づきコンパイラで生成された、オリジナルのシンボル名がそのまま用いられる。図 5 の例では、ソースファイル 2 0 における「func1(obj)」に対して、オブジェクトファイル 2 1 中の「_func1_6Class1P6Class1」がシンボル名として用いられる。シンボル名に対して、参照先のコンポーネントにおける対応する、相対的なアドレス情報が付加される。

【 0 0 3 9 】

また、シンボル参照情報テーブル 2 2 と、上述の参照先コンポーネント 4 B を生成する際に作成された外部シンボル情報ファイル 1 3 とから、参照先コンポー

メント4 Bについての外部シンボル情報ファイル13' が作成される(図5 B)。外部シンボル情報ファイル13と、シンボル参照テーブル22とに基づき、オリジナルのシンボル名に対応するコンポーネントIDとシンボルIDとが得られる。得られたコンポーネントIDとシンボルIDとの対と、オリジナルのシンボル名とを対応付ける。これを、シンボル参照情報テーブル22に列挙されるオリジナルのシンボル名について、全て行う。

【0040】

オブジェクトファイル21と外部シンボル情報ファイル13' とから、参照元のコンポーネント4 Aが生成される(図5 C)。コンポーネント4 Aは、参照先のコンポーネントIDおよびシンボルIDの対と、シンボルの参照アドレスとが対応付けられて列挙された、シンボル参照情報テーブル23を有する。図5 Cの例では、コンポーネントIDとシンボルIDの対は、「C45__S3」のように表現され、これがシンボルの参照場所を示すシンボル値と対応付けられる。

【0041】

シンボル参照情報テーブル23は、外部シンボル情報ファイル13' を用いて、オブジェクトファイル21のシンボル参照情報テーブル22におけるオリジナルのシンボル名をコンポーネントIDとシンボルIDとの対に変換したものであり、置き換えられたシンボル名と、コンポーネント内の、シンボルを参照しているアドレス情報を有している。

【0042】

すなわち、シンボル参照情報テーブル23は、オブジェクトファイル21中のシンボル参照情報テーブル22におけるオリジナルのシンボル名が、コンポーネントIDとシンボルIDの対からなる新たなシンボル名によって置き換えられたと考えることができる。

【0043】

なお、ここでは、他のコンポーネントから参照されるシンボルを持つコンポーネントと、他のコンポーネントが持つシンボルを参照するコンポーネントを別々に説明しているが、これはこの例に限られない。一つのコンポーネントが他のコンポーネントから参照されるシンボルを持ち、且つ、そのコンポーネントが他の

コンポーネントが持つシンボルを参照することも可能であり、そのような場合でも、この発明を適用することができる。

【 0 0 4 4 】

次に、上述のようにして生成された参照元コンポーネント 4 A と参照先コンポーネント 4 B とをリンクして実行ファイルを作成する方法について説明する。図 6 は、ターゲットシステム 1 上でのコンポーネントのリンク機構を概略的に示す。ここでは、ターゲットシステム 1 では予め所定のプログラムが起動され、あるコンポーネント 4 A が RAM 1 3 上に読み込まれているものとする。このコンポーネント 4 A の記述に基づき、他のコンポーネント 4 B が呼び出され、コンポーネント 4 A に対してリンクされる。なお、これは、実際には、OS 2 0 の制御に基づきなされる処理である。

【 0 0 4 5 】

コンポーネント 4 A のシンボル参照情報テーブル 2 3 に基づき、コンポーネント 4 A から OS 3 0 に対して、コンポーネント 4 B のリンクが要求される。このリンク要求は、OS 3 0 からリンカ 3 1 に渡される。リンカ 3 1 は、この要求に基づき参照元のコンポーネントのシンボル参照情報テーブル 2 3 と、参照先のコンポーネント 4 B のシンボル情報テーブル 1 4 とを参照する。そして、リンカ 3 1 は、これらテーブル 2 3 および 1 4 に記述されているコンポーネント ID およびシンボル ID に基づき、参照元コンポーネント 4 A と参照先コンポーネント 4 B とをリンクする。

【 0 0 4 6 】

図 7 は、ターゲットシステム 1 におけるコンポーネント間のリンクを行う一例の手順を示すフローチャートである。このフローチャートによる処理は、OS 3 0 の制御に基づき、リンカ 3 1 によってなされるものである。最初のステップ S 1 0 で、参照元のコンポーネント 4 A と参照先のコンポーネント 4 B とが RAM 1 3 に読み込まれ、ターゲットシステム 1 にロードされる。これは、例えば図示されないコンポーネントローダによって、参照元コンポーネント 4 A に記述された、参照先のコンポーネント 4 B のコンポーネント名やバージョン情報に基づき、2 次記憶媒体 3 に格納されているコンポーネントから適切なコンポーネントが検

索されて、なされる。

【0047】

参照元および参照先のコンポーネント4 Aおよび4 Bがそれぞれロードされたら、上述したように、参照元のコンポーネント4 AからOS 30に対して参照元コンポーネント4 Bに対するリンク要求が出され、このリンク要求がOS 30からリンカ31に渡される。次のステップS11で、リンカ31によって、参照元のコンポーネント4 Aのシンボル参照情報テーブル23から、エントリが一つ取得される。ここで、シンボル参照情報テーブル23の全てのエントリに対しての処理が終了していると判断されたなら（ステップS12）、一連のフローチャートが終了される。

【0048】

若し、ステップS12で、参照元情報テーブル23において処理が終了していないエントリが存在するとされたら、処理はステップS13に移行する。ステップS13では、参照元コンポーネント4 Aのシンボル参照情報テーブル23に基づき、参照先コンポーネント4 Bのシンボル情報テーブル14の該当するエントリが検索される。上述したように、参照元コンポーネント4 Aのシンボル参照情報テーブル23におけるシンボル名は、コンポーネントIDとシンボルIDとの組み合わせで構成されているので、シンボル参照情報テーブル23の記述に基づき、参照先のコンポーネント4 BのコンポーネントIDが取得される。

【0049】

ステップS14では、取得されたコンポーネントIDが、リンク先である参照先コンポーネント4 BのコンポーネントIDと一致しているかどうか判断される。若し、一致していないとされれば、処理はステップS17に移行し、参照先コンポーネント4 Bに対して参照元コンポーネント4 Aをリンクしないままにしておく。そして、処理はステップS11に戻され、シンボル参照情報テーブル23の次のエントリが取得される。

【0050】

一方、ステップS14で、ステップS13で取得されたコンポーネントIDが参照先コンポーネント4 BのコンポーネントIDと一致していると判断されれば

、処理は次のステップ S 1 5 に移行する。ステップ S 1 5 では、リンク先となる参照先コンポーネント 4 B のシンボル情報テーブル 1 4 が参照され、リンクの対象となるシンボル情報エントリが取得される。

【 0 0 5 1 】

すなわち、ステップ S 1 5 では、シンボル参照情報テーブル 2 3 に基づき、ステップ S 1 3 で取得されたコンポーネント I D と対とされているシンボル I D が求められる。そして、求められたシンボル I D に基づき、参照先コンポーネント 4 B のシンボル情報テーブル 1 4 のエントリが検索される。

【 0 0 5 2 】

ステップ S 1 5 では、上述のステップ S 1 4 の検索結果に基づき、シンボル I D に対応するシンボル値が求められる。これにより、シンボル参照情報テーブル 2 3 に記述されている参照場所の参照が解決され、シンボルのリンクが行われる（ステップ S 1 6）。これら一連の処理が、参照元コンポーネント 4 A のシンボル参照テーブル 2 3 の各エントリに対して行われる。

【 0 0 5 3 】

なお、参照先のコンポーネント 4 B の元となるオブジェクトファイル 1 1 に用いられたオリジナルのシンボル名と、参照元コンポーネント 4 の元となるオブジェクトファイル 2 1 に用いられたオリジナルのシンボル名は、これらのコンポーネントの生成時には、全て、コンポーネント I D とシンボル I D との組み合わせによるシンボル名に置き換えられている。そのため、参照先のコンポーネントの外部シンボル参照ファイル 1 3 および参照元のコンポーネントの外部シンボル参照ファイル 1 3' 、ならびに、各シンボル名を置き換える以前の、オリジナルのシンボル名に関する情報は、コンポーネント生成以降は、不要となる。

【 0 0 5 4 】

【発明の効果】

以上説明したように、この発明によれば、ターゲットシステム 1 上で必要とされるシンボル情報は、シンボル I D に置き換えられて用いられる。シンボル I D は、オリジナルのシンボル名に比べて短くすることが可能である。これにより、シンボル名をターゲットシステム 1 上に格納するためのメモリの量を軽減するこ

とができると共に、コンポーネントのリンク時などにおけるシンボル検索の際の処理時間を短くすることができるという効果がある。

【 0 0 5 5 】

また、この発明によれば、参照元のシンボル参照情報にコンポーネント ID を付加している。これにより、コンポーネントのリンク時に、シンボル参照情報の参照先のシンボルがリンク先のコンポーネントに存在するかどうかを、コンポーネント ID を比較するだけで判断することができる。したがって、シンボルの検索に要する処理時間を短縮することができるという効果がある。

【 0 0 5 6 】

さらに、この発明によれば、全てのコンポーネントのシンボル名が置き換えられたものとして扱われ、また、シンボル ID に対してそれぞれのシンボルが属しているコンポーネントの識別情報（コンポーネント ID）が付加されるため、異なる 2 つ以上のコンポーネントにおいて、同一名称のシンボル名が宣言された場合でも、これらを別々のシンボルとして扱うことができるという効果がある。

【図面の簡単な説明】

【図 1】

この発明の実施の一形態に適用可能なシステムの概念を示す略線図である。

【図 2】

ターゲットシステムが構成される情報機器の一例の構成を示すブロック図である。

【図 3】

リンク先となるコンポーネントの一例の生成方法を示す略線図である。

【図 4】

外部シンボル情報ファイルの一例の構成を示す略線図である。

【図 5】

リンク元となるコンポーネントの一例の生成方法を示す略線図である。

【図 6】

ターゲットシステム上でのコンポーネントのリンク機構を概略的に示すブロック図である。

【図 7】

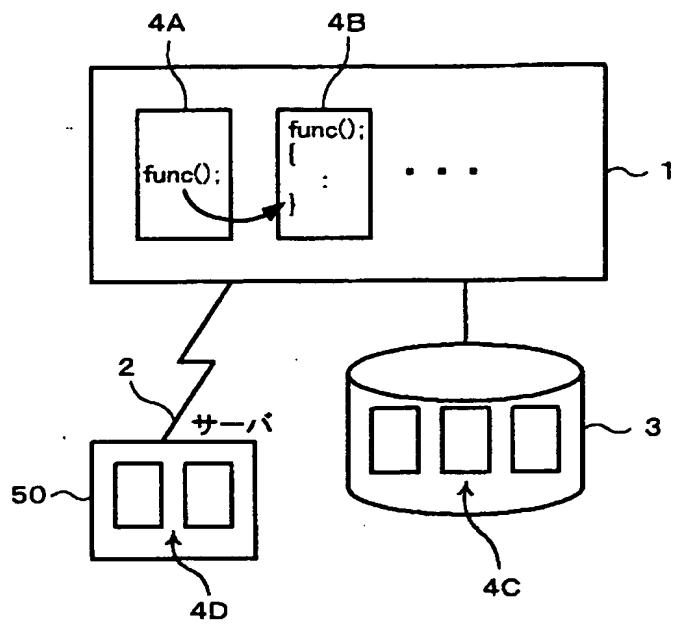
ターゲットシステムにおけるコンポーネント間のリンクを行う一例の手順を示すフローチャートである。

【符号の説明】

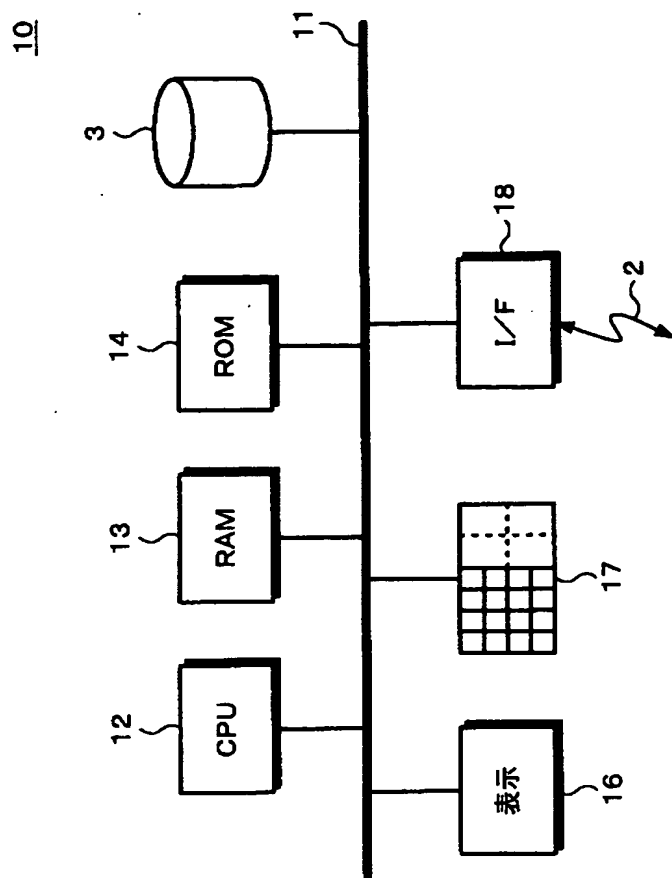
1 . . . ターゲットシステム、2 . . . ネットワーク、3 . . . 2 次記憶装置、
4 A . . . 参照元のコンポーネント、4 B . . . 参照先のコンポーネント、1 3
, 1 3' . . . 外部シンボル参照ファイル、1 4 . . . シンボル情報テーブル、
2 3 . . . シンボル参照情報テーブル

【書類名】 図面

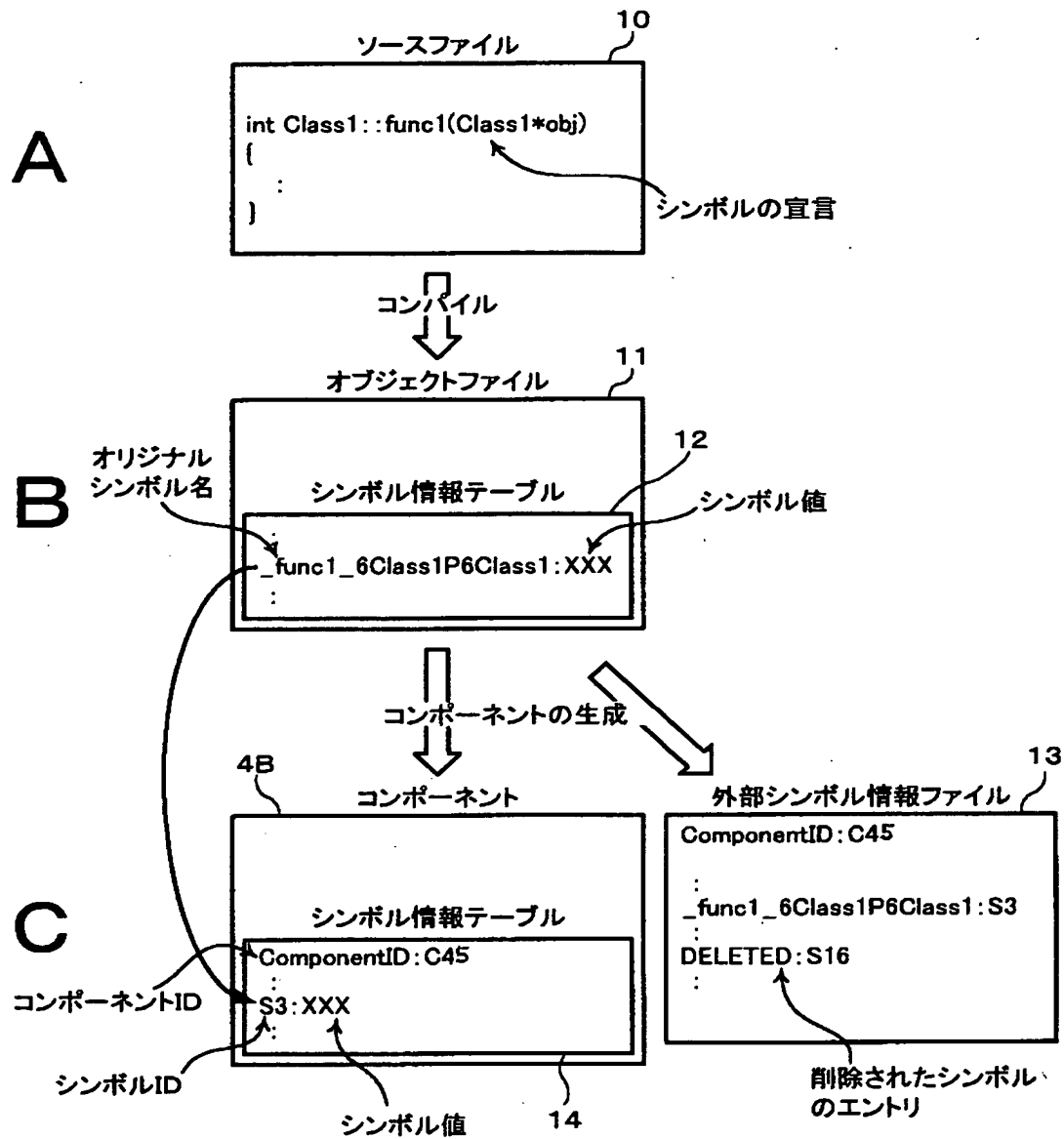
【図 1】



【図 2】



【図 3】

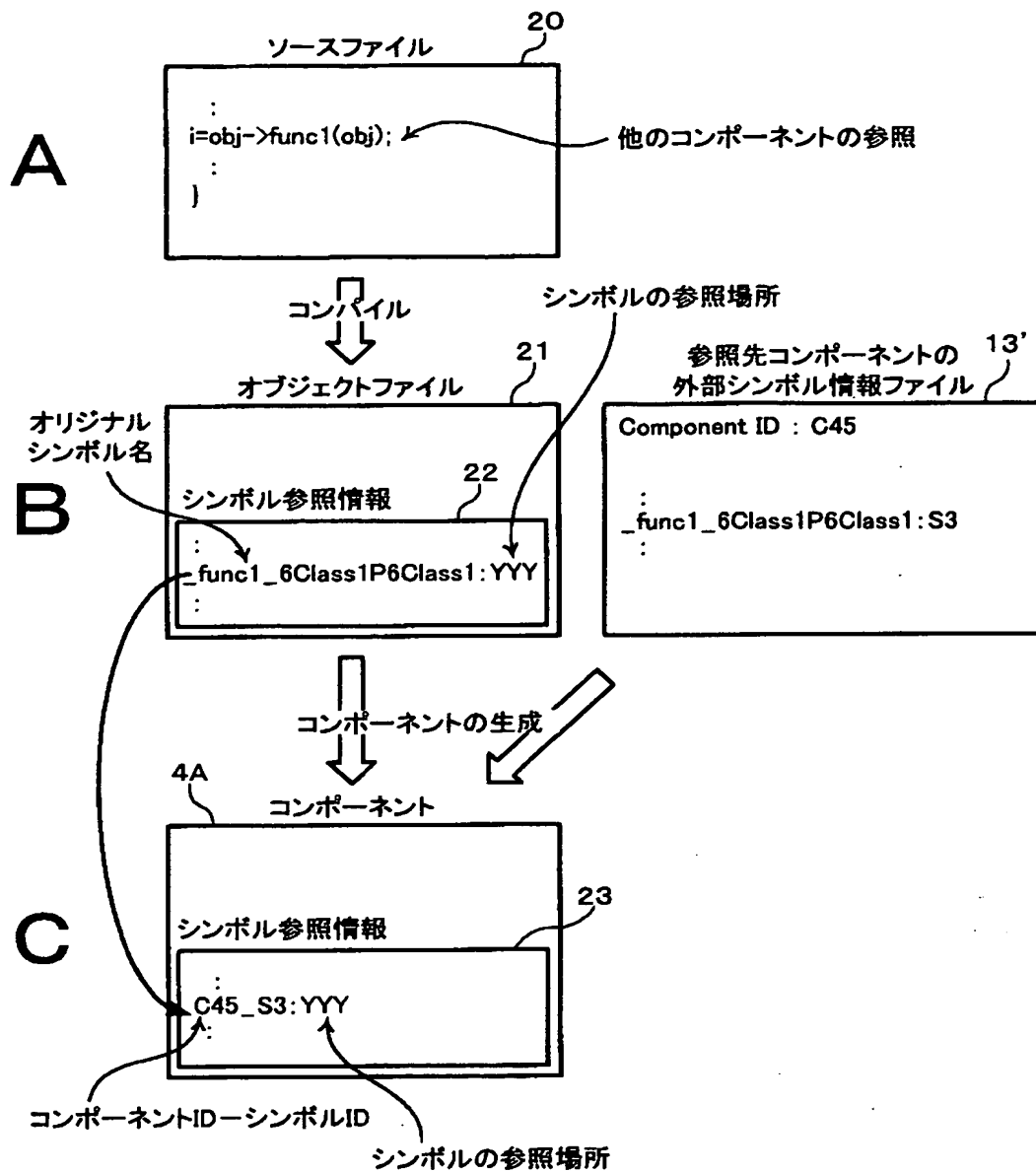


【図 4】

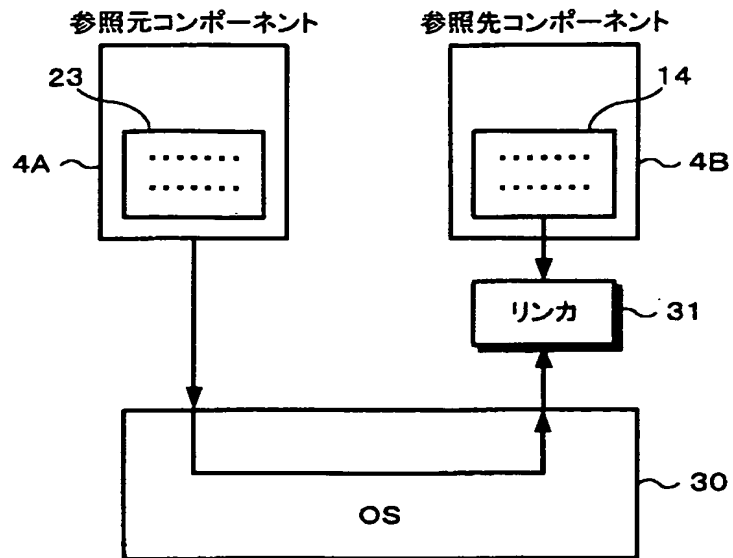
13

ComponentID	コンポーネントID
オリジナルシンボル名1	シンボルID1
オリジナルシンボル名2	シンボルID2
⋮	⋮
オリジナルシンボル名n	シンボルIDn

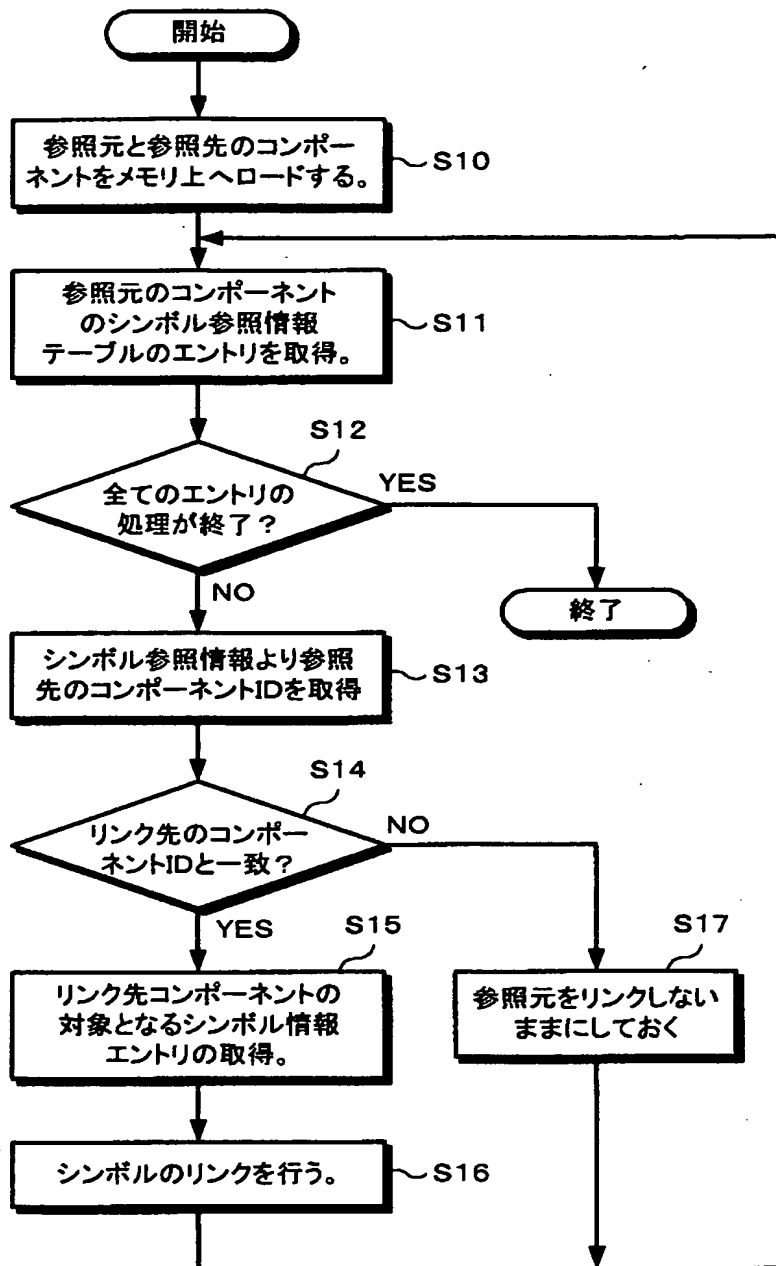
【図 5】



【図 6】



【図 7】



【書類名】 要約書

【要約】

【課題】 コンポーネントのリンクに要する処理時間を短縮すると共に、リンク時のメモリの使用量が少なく抑える。

【解決手段】 他のコンポーネントから参照されるコンポーネントの作成に際し、コンパイル処理によりソースファイル 1 0 中のシンボルの宣言がオリジナルシンボル名に変換され、オブジェクトファイル 1 1 内のオフセットアドレスであるシンボル値と対応付けられる。参照先コンポーネント 4 B の作成時に、オリジナルシンボル名がシンボル I D に置き換えられてシンボル値と対応付けられたシンボル情報テーブル 1 4 が作成される。また、オリジナルシンボル名とシンボル I D とが対応され外部シンボル情報ファイル 1 3 が作成される。参照元コンポーネントでは、ファイル 1 3 を用いて、他のコンポーネントを参照するオリジナルシンボル名がコンポーネント I D とシンボル I D との対に置き換えられ、シンボル参照情報テーブル 2 3 が作成される。参照元および参照先コンポーネントのリンクは、テーブル 2 3 とテーブル 1 4 とに基づき、シンボル I D を用いてリンク情報の照合が行われる。シンボル I D は短いので、シンボルの照合に要する時間が短縮されると共に、メモリの使用量が少なくできる。

【選択図】 図 3

出 願 人 履 歴 情 報

識別番号 [000002185]

1. 変更年月日	1990年 8月30日
[変更理由]	新規登録
住 所	東京都品川区北品川6丁目7番35号
氏 名	ソニー株式会社